Дискретные модели данных в компьютере. Представление чисел.

Презентация для 10 класса

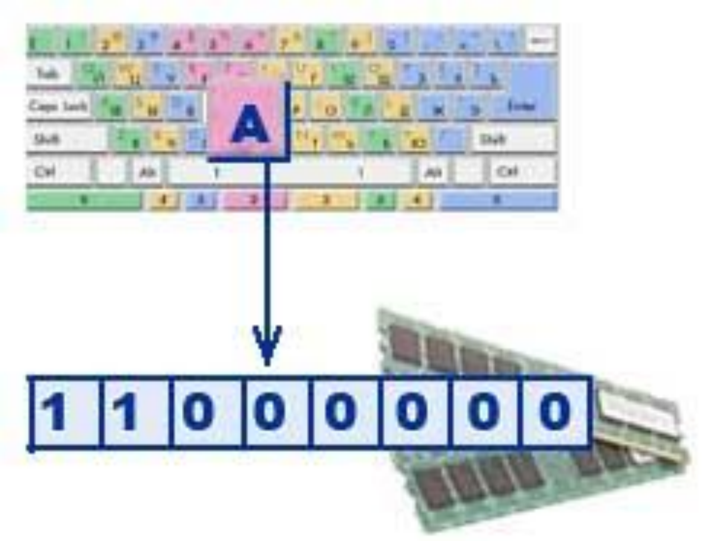
Образ компьютерной памяти

1	1	0	0	1	1	1	1	0	0	0	1	0	1	0	0
0	1	0	0	1	1	1	0	1	0	0	0	1	1	1	1
1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	0	1	1	0	1	1	1
1	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0
1	1	1	0	0	1	0	1	0	1	1	1	0	0	0	1
1	0	1	0	1	0	1	0	1	1	1	1	1	0	0	0
1	1	0	1	0	1	0	1	1	0	0	1	1	1	0	0
0	1	0	1	0	0	1	1	1	1	1	0	0	0	1	1
1	0	0	1	1	0	0	0	1	1	1	0	1	0	1	0
1	0	0	0	1	1	0	1	0	1	0	1	0	1	0	0
1	1	0	0	1	1	1	0	0	1	1	0	1	0	1	1
0	0	1	1	1	0	0	1	1	0	1	0	1	0	1	0
0	1	1	1	1	0	1	0	1	0	1	0	0	1	1	1
0	0	1	1	1	0	1	0	1	0	1	0	1	1	1	0
1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	1

Главные правила представления данных в компьютере

Правило № 1

Данные (и программы) в памяти компьютера хранятся в двоичном виде, т.е. в виде цепочек единиц и нулей.



Правило № 2

Представление данных в компьютер дискретно.

Дискретизация — преобразование непрерывной функции в дискретную.

- Дискретность (от лат. discretus — **разделённый, прерывистый**), прерывность; противопоставляется непрерывности. Например, дискретное изменение какой-либо величины во времени — это изменение, происходящее через определённые промежутки времени (скачками); система целых чисел (в противоположность системе действительных чисел) является дискретной . В физике и химии Д. означает зернистость строения материи, её атомистичность.

ДИСКРЕТНОСТЬ [discretion] — **прерывность**; напр., изменение экономических показателей во времени всегда имеет прерывный характер, поскольку происходит скачками — от одной даты (года, месяца и т. д.) к другой. Понятие Д. противопоставляется понятию непрерывности.

Правило № 3

Множество представленных в памяти величин ограничено и конечно.

символ	10-й код	2-й код	символ	10-й код	2-й код
	32	00100000	8	56	00111000
!	33	00100001	9	57	00111001
"	34	00100010	:	58	00111010
#	35	00100011	;	59	00111011
$	36	00100100	<	60	00111100
%	37	00100101	=	61	00111101
&	38	00100110	>	62	00111110
'	39	00100111	?	63	00111111
(	40	00101000	@	64	01000000
)	41	00101001	A	65	01000001
*	42	00101010	B	66	01000010
+	43	00101011	C	67	01000011
,	44	00101100	D	68	01000100

Представление чисел в ПК

Целые числа в компьютере

Правило № 4

В памяти компьютера числа хранятся в двоичной системе счисления.

Например, если под целое число выделяется ячейка памяти размером в 16 битов, то самое большое целое положительное число будет таким:

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

В десятичной системе счисления оно равно:

$$2^{15} - 1 = 32\,767.$$

Представление чисел в формате с фиксированной запятой

Целые числа в компьютере хранятся в памяти в формате *с фиксированной запятой*. В этом случае каждому разряду ячейки памяти соответствует всегда один и тот же разряд числа, а запятая находится справа после младшего разряда, т.е. вне разрядной сетки.

Для хранения *целых неотрицательных чисел* отводится одна ячейка памяти (8 бит). Например, число $A_2 = 10101010_2$ будет хранится в ячейке памяти следующим образом:

1	0	1	0	1	0	1	0

Максимальное значение целого неотрицательного числа достигается в случае, когда во всех ячейках хранятся единицы. Для n-разрядного представления оно будет равно:

$$2^n - 1$$

Пример. Определить диапазон чисел, которые могут хранится в оперативной памяти в формате *целое неотрицательное число*.

Минимальное число соответствует восьми нулям, хранящимся в восьми ячейках памяти, и равно нулю.

Максимальное число соответствует восьми единицам, хранящимся в ячейках памяти и равно:

$A = 1*2^7 +1*2^6 +1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 1*2^0 = 1*2^8 – 1 = 255_{10}$

Диапазон изменения *целых неотрицательных чисел* от 0 до 255.

Для хранения ***<u>целых чисел со знаком</u>*** отводится две ячейки памяти (16 бит), причем старший (левый) разряд отводится под знак числа (если число положительное, то в знаковый разряд записывается **0**, если число отрицательное записывается **1**).

Представление в компьютере положительных чисел с использованием формата «знак-величина» называется ***<u>прямым кодом</u>*** числа.

Например, число $2002_{10} = 11111010010_2$ будет представлено в 16-ти разрядном представлении следующим образом:

0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0

При представлении целых чисел в n-разрядном представлении со знаком максимальное положительное число (с учетом выделения одного разряда на знак) равно:

$A = 2^{n-1} - 1$

Пример. Определить максимальное положительное число, которое может хранится в оперативной памяти в формате *целое число со знаком*.

$A_{10} = 2^{15} - 1 = 32767_{10}$

Для представления отрицательных чисел используется ***дополнительный код***. Дополнительный код позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно упрощает работу процессора и увеличивает его быстродействие.

Дополнительный код отрицательного числа А, хранящегося в *n* ячейках, равен 2^n - |А|.

Дополнительный код представляет собой дополнение модуля отрицательного числа А до 0, поэтому в n-разрядной компьютерной арифметике:

$$2^n - |A| + |A| \equiv 0$$

Это равенство тождественно справедливо, т.к. в компьютерной n-разрядной арифметике $2^n \equiv 0$. Действительно, двоичная запись такого числа состоит из одной единицы и *n* нулей, а в n-разрядную ячейку может уместиться только n младших разрядов, т.е. *n* нулей.

ПРИМЕР. ЗАПИСАТЬ ДОПОЛНИТЕЛЬНЫЙ КОД ОТРИЦАТЕЛЬНОГО ЧИСЛА –2002 ДЛЯ 16-ТИ РАЗРЯДНОГО КОМПЬЮТЕРНОГО ПРЕДСТАВЛЕНИЯ

Проведем вычисления в соответствии с определением дополнительного кода:

2^{16}	=	10000000000000000_2	65536_{10}
2002_{10}	=	0000011111010010_2	2002_{10}
2^{16} - $\|2002_{10}\|$	=	1111100000101110_2	63534_{10}

Проведем проверку с использованием десятичной системы счисления. Дополнительный код 63534_{10} в сумме с модулем отрицательного числа 2002_{10} равен 65536_{10}, т.е. дополнительный код дополняет модуль отрицательного числа до 2^{16} (до нуля 16-ти разрядной компьютерной арифметики).

Для получения дополнительного кода отрицательного числа можно использовать довольно простой алгоритм:

ПРАВИЛО ПОЛУЧЕНИЯ ДОПОЛНИТЕЛЬНОГО КОДА

Для получения дополнительного кода отрицательного числа можно использовать довольно простой алгоритм:

1. Модуль числа записать ***прямым кодом*** в *n* двоичных разрядах;
2. Получить ***обратный код*** числа, для этого значения всех бит инвертировать (все единицы заменить на нули и все нули заменить на единицы);
3. К полученному ***обратному коду*** прибавить **единицу.**

ПРИМЕР ЗАПИСАТЬ ДОПОЛНИТЕЛЬНЫЙ КОД ОТРИЦАТЕЛЬНОГО ЧИСЛА –2002 ДЛЯ 16-ТИ РАЗРЯДНОГО КОМПЬЮТЕРНОГО ПРЕДСТАВЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА.

Прямой код	$\lvert -2002_{10} \rvert$	0000011111010010_2
Обратный код	инвертирование	1111100000101101_2
	прибавление единицы	1111100000101101_2 $+\ 0000000000000001_2$
Дополнительный код		1111100000101110_2

При n-разрядном представлении отрицательного числа А дополнительным кодом старший разряд выделяется для хранения знака числа (единицы). В остальных разрядах записывается положительное число:

$$\mathbf{2^{n-1} - |A|.}$$

Чтобы число было положительным должно выполняться условие:

$$\mathbf{|A| \leq 2^{n-1}}$$

Следовательно, максимальное значение модуля числа А в n-разрядном представлении равно:

$$\mathbf{|A| = 2^{n-1}}$$

Тогда, минимальное отрицательное число равно:

$$\mathbf{A = -2^{n-1}}$$

Пример. Выполнить арифметическое действие 3000_{10} - 5000_{10} в 16-ти разрядном компьютерном представлении.

Представим положительное число в прямом, а отрицательное число в дополнительном коде:

Десятичное число	Прямой код	Обратный код	Дополнительный код
3000	0000101110111000		
-5000	0001001110001000	1110110001110111	1110110001110111 +0000000000000001 1110110001111000

СЛОЖИМ ПРЯМОЙ КОД ПОЛОЖИТЕЛЬНОГО ЧИСЛА С ДОПОЛНИТЕЛЬНЫМ КОДОМ ОТРИЦАТЕЛЬНОГО ЧИСЛА. ПОЛУЧИМ РЕЗУЛЬТАТ В ДОПОЛНИТЕЛЬНОМ КОДЕ:

3000-5000			1111100000110000

Переведем полученный дополнительный код в десятичное число:

1) Инвертируем дополнительный код: 0000011111001111

2) Прибавим к полученному коду 1 и получим модуль отрицательного числа:

$$\begin{array}{r} 0000011111001111 \\ +\ \underline{0000000000000001} \\ 0000011111010000 \end{array}$$

3) Переведем в десятичное число и припишем знак отрицательного числа: -2000.

Недостатком представления чисел в формате *с* ***фиксированной запятой*** является конечный диапазон представления величин, недостаточный для решения математических, физических, экономических и других задач, в которых используются как очень малые, так и очень большие числа.

Вывод:

Целые числа в памяти компьютера – это дискретное, ограниченное и конечное множество.

Границы множества целых чисел зависят от размера выделяемой ячейки памяти под целое число, а также от формата: со знаком или без знака.

МАТЕМАТИКА:
множество целых чисел дискретно, бесконечно, не ограничено

ИНФОРМАТИКА:
множество целых чисел дискретно, конечно, ограничено

Границы множества целых чисел зависят от размера выделяемой ячейки памяти под целое число, а также от формата: со знаком или без знака.